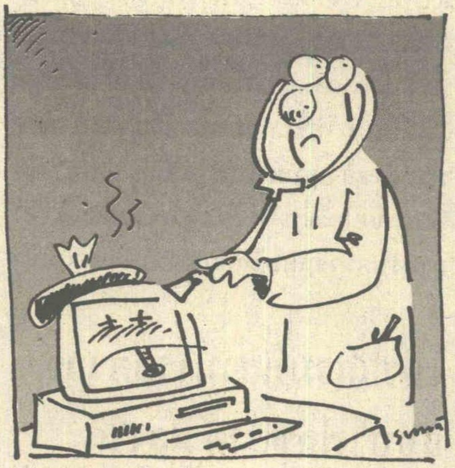
จเร เลิศสุดวิชัย

เขียนโปรแกรม เช็คไวรัส ด้วยตนเอง

เสนอแนวทางในการพัฒนาโปรแกรมให้สามารถตรวจสอบการติดหรือแพร่กระจายได้ด้วยตนเอง



ภาพประกอบ : สุชาติ สหชาติโภคานันท์

ปัจจุบันไวรัสคอมพิวเตอร์ได้สร้างความวุ่นวายและความเดือดร้อนทั่วไปในวงการคอมพิวเตอร์ ถึงจะมีผู้พัฒนาและสร้างโปรแกรมปราบไวรัสออกมาต่อสู้อยู่เสมอก็ตาม แต่ก็ยังไม่มีโปรแกรมที่จะมีความสามารถตรวจสอบและปราบปรามไวรัสได้ครอบคลุมหมดทุกตัว โดยเฉพาะเจ้าตัวไวรัสเองก็ไม่ได้หยุดพักเช่นกัน จะพบว่ามีไวรัสพันธุ์ใหม่ ๆ ปรากฏให้พบเห็นอยู่ตลอดเวลา ลำพังไวรัสพันธุ์เก่า ๆ เองก็สร้างปัญหาให้มากพออยู่แล้ว แต่ยังต้องมาผจญกับไวรัสพันธุ์ใหม่ที่เกิดขึ้นอยู่ตลอดเวลาอีก เราจะหาหนทางหรือวิธีการป้องกันการแพร่กระจายไวรัสอย่างไรดี หลาย ๆ ท่านคงคิดหาหนทางกันอยู่แล้ว แต่สำหรับท่านที่กำลังเขียนโปรแกรมภาษาซีอยู่ในขณะนี้ ในมุมภาษาซีฉบับนี้จะได้เสนอแนวทางในการพัฒนาโปรแกรมของท่านให้สามารถตรวจสอบการติดหรือการแพร่กระจายได้ด้วยตัวเอง ซึ่งผู้เขียนคิดว่าคงจะสร้างประโยชน์ให้กับท่านได้มากทีเดียว

ไวรัสคอมพิวเตอร์ประเภทไฟล์

วิธีการที่ไวรัสจะแพร่กระจายได้รวดเร็วและเพิ่มจำนวนได้มากที่สุด คือ วิธีการเกาะติดไปกับไฟล์ที่เป็นโปรแกรม ซึ่งวิธีนี้เป็นวิธีที่ไวรัสคอมพิวเตอร์ในปัจจุบันนิยมใช้มากที่สุด การแพร่กระจายวิธีนี้จะเกิดขึ้นเมื่อไวรัสได้ทำงานอยู่ในหน่วยความจำของเครื่องแล้ว ถ้ามีการโหลดไฟล์ที่เป็นโปรแกรม (ซึ่งได้แก่ไฟล์พวกนามสกุล .COM และ .EXE) โดยดอสโปรแกรมไวรัสจะทำการแพร่กระจายทันที โดยจะทำการเขียนไฟล์โปรแกรมที่ถูกโหลดขึ้นลงดิสค์ ซึ่งตัวไฟล์โปรแกรมจะได้รับการดัดแปลงใหม่ โดยมีโปรแกรมไวรัสเกาะติดอยู่ในส่วนท้ายหรือส่วนหัวของไฟล์ ไฟล์ดังกล่าวที่ถูกดัดแปลงนั้นจะทำหน้าที่เป็นตัวแพร่กระจายอีกต่อไป ซึ่งเมื่อมีการเรียกไฟล์ดังกล่าวขึ้นมาทำงานเมื่อใดไวรัสที่เกาะติดอยู่ก็จะถูกปลุกขึ้นมาด้วยทุกครั้งไป

การดัดแปลงไฟล์โปรแกรมดังกล่าว นอกจากการนำเอาตัวโปรแกรมไวรัสไปเกาะติดในไฟล์แล้ว ยังมีการดัดแปลงโครงสร้างการเริ่มต้นทำงานของโปรแกรมใหม่ด้วย ซึ่งในกรณีของไฟล์ชนิด .EXE

จะมีเนื้อไฟล์ส่วนหนึ่งตอนหัวซึ่งเรียกว่า เฮดเดอร์ นั้นใช้เก็บข้อมูลบางอย่างที่ต้องใช้ในการโหลดโปรแกรมลงสู่หน่วยความจำ

แนวความคิดในการตรวจสอบการติดไวรัสของไฟล์

จากการแพร่กระจายของไวรัสดังกล่าวนั้น เราจะพบว่ามีวิธีการตรวจสอบง่าย ๆ ว่าไฟล์โปรแกรมใดที่ติดไวรัสแล้ว ซึ่งการตรวจสอบก็จะตรวจสอบได้จากขนาดของไฟล์ที่เปลี่ยนไปนั้นเอง การตรวจสอบขนาดไฟล์อย่างเดียวอาจจะเป็นวิธีตรงและง่ายแต่วิธีนี้วิธีเดียวเพียงพอแล้วหรือ ยังหรอกครับไวรัสในปัจจุบันอาจจะใช้วิธีการง่าย ๆ เที่ยวเกาะติดไปกับไฟล์เท่านั้น ซึ่งจะทำให้ขนาดของไฟล์เพิ่มขึ้นสังเกตเห็นได้ชัดเจน แต่ในอนาคตโปรแกรมไวรัสอาจจะฉลาดขึ้นแทนที่จะเอาตัวเองไปเกาะติดไฟล์เฉย ๆ อาจหันไปใช้วิธีลดขนาด (compress) ของไฟล์ที่จะเกาะให้เล็กลงก่อนแล้วนำตัวเองไปเกาะพร้อมกับเขียนอะไรต่ออีกเพื่อให้ไฟล์มีขนาดเท่าเดิม ในกรณีนี้การตรวจสอบขนาดอย่างเดียวคงทำไม่ได้แน่ ถ้าอย่างนั้นทำอย่างไรดีล่ะครับถึงตรวจสอบได้ครอบคลุมหมด ใช่ครับต้องลงตรวจละเอียดถึงเนื้อไฟล์เลยทีเดียว หลาย ๆ ท่านคงคิดถึงวิธีการตรวจสอบผลรวม หรือ checksum (ซึ่งหมายถึงการนำเอาค่าในเนื้อไฟล์แต่ละไบต์บวกเข้าด้วยกันจนหมด สุดท้ายจะได้ผลลัพธ์ตัวเลขตัวเลขหนึ่ง) ซึ่งเป็นวิธีที่ง่ายแต่โอกาสจะผิดพลาดก็มีมากเนื่องจากตัวเลขที่ได้มีโอกาสซ้ำกัน วิธีการที่จะช่วยลดความผิดพลาดให้น้อยลงคือวิธี CRC (Cyclic redundancy check) วิธีการของ CRC นั้นจะคล้ายกับ Check Sum (แต่ไม่ได้ใช้วิธีการบวกธรรมดา) คือสร้างตัวเลขขึ้นมาหนึ่งจำนวนจากเนื้อไฟล์ทั้งหมด

การตรวจสอบ CRC ของไฟล์นั้นถ้าจำไม่ผิดในโปรแกรมชุดของ VIRUSCAN ของไอบีเอ็มจะมีโปรแกรม CRC.COM แถมมา ซึ่งโปรแกรมจะทำงานในครั้งแรกโดยการอ่านไฟล์ทุกไฟล์ที่เป็นโปรแกรมเพื่อนำมาหาตัวเลขประจำของแต่ละไฟล์โดยวิธี CRC ซึ่งตัวเลขและชื่อไฟล์จะถูกจัดเก็บเอาไว้สำหรับตรวจสอบไฟล์เหล่านั้นอีกภายหลัง

โปรแกรมที่สามารถตรวจสอบไฟล์ของตัวเองได้

ท่านที่เคยใช้โปรแกรม SCAN.EXE รุ่นใหม่ ๆ จะพบตัวโปรแกรม SCAN สามารถทำการตรวจสอบไฟล์ตัวเองก่อนทำงานอื่น และรายงานผลเมื่อตัวโปรแกรมเกิดติดไวรัสได้ด้วย การทำงานของโปรแกรมลักษณะนี้แหละคือสิ่งที่เราจะพูดถึงในมุมภาษาซีฉบับนี้

โปรแกรมที่เราจะสร้างขึ้นนั้นก่อนที่จะทำงานอื่นใด จะต้องอ่านเอาไฟล์ของตัวเองขึ้นมาจากดิสค์และทำการหาขนาดไฟล์และค่าตัวเลขที่ได้จากการคำนวณ CRC แล้วนำค่านั้นมาตรวจสอบกับค่าที่เก็บเอาไว้ ถ้าขนาดไฟล์หรือค่าตัวเลข CRC ไม่ตรงกันก็รายงานแสดงข้อความปรากฏบนจอภาพให้เห็นว่าไฟล์โปรแกรมมีการเปลี่ยนแปลงซึ่งอาจจะเจอไวรัสเข้าแล้ว

แต่เอ๊ะ! แล้วตัวโปรแกรมจะรู้ได้อย่างไรล่ะครับว่า โปรแกรมที่เราเขียนขึ้นมานี้มีขนาดเท่าไร และค่า CRC เป็นเท่าไร ขณะที่เราเขียนโปรแกรมอยู่จะรู้ขนาดของไฟล์ และค่า CRC ทันทีเลยหรือ แน่นอนครับไม่มีทางที่จะรู้ค่าเหล่านั้นได้ก่อนที่จะเขียนโปรแกรมเสร็จ แล้วจะใช้ค่าที่ไหนเป็นค่าสำหรับเปรียบเทียบกับค่าที่จะคำนวณได้เมื่อโปรแกรมเริ่มทำงาน

ในโปรแกรมที่จะพัฒนาขึ้นจะประกอบด้วยโมดูลพิเศษที่เพิ่มเข้าไปนอกเหนือไปจากโปรแกรมปกติที่พัฒนาขึ้น โมดูลนี้จะถูกเรียกขึ้นมาทำงานก่อนทันทีหลังจากโปรแกรมถูกโหลดลงหน่วยความจำ ซึ่งการทำงานในโมดูลนี้จะคำนวณหาค่าขนาดไฟล์ของโปรแกรมและค่า CRC จากเนื้อไฟล์โปรแกรมและจากค่าที่คำนวณได้ จะนำมาเปรียบเทียบกับค่าขนาดไฟล์และค่า CRC ที่เก็บไว้ซึ่งผลการเปรียบเทียบพบว่าไม่มีการเปลี่ยนแปลงทั้งขนาดไฟล์และค่า CRC ก็ผ่านไปทำงานตามปกติ แต่ถ้ามีการเปลี่ยนแปลงอย่างใดอย่างหนึ่งหรือทั้งสองค่าก็จะพิมพ์ข้อความแสดงผลลัพธ์ออกมาว่าเกิดการเปลี่ยนแปลงขึ้นในไฟล์โปรแกรม

คราวนี้มาถึงปัญหาเรื่องค่าขนาดของไฟล์และค่า CRC ของไฟล์ที่ต้องใช้เก็บไว้ใช้ในการเปรียบเทียบ ตามที่กล่าวมาแล้วในขณะเขียนโปรแกรมเราไม่สามารถจะรู้ล่วงหน้าถึงค่าขนาดไฟล์และค่า CRC ของไฟล์ได้เลย ค่าขนาดไฟล์และค่า CRC จะถูกกำหนดเป็นตัวแปรอยู่ในส่วนโมดูลของการตรวจสอบ ซึ่งจะขอเรียกโมดูลส่วนนี้ว่า EXEAWARE ตัวแปรที่ใช้สำหรับเก็บค่าขนาดไฟล์และค่า CRC ดังกล่าวนั้นตอนเริ่มเขียนโปรแกรมไม่จำเป็นต้องกำหนดค่าใด ๆ ให้เลยหรือจะใส่ค่าศูนย์ลงไปในตัวแปรทั้งสองก็ได้ แต่การกำหนดค่าจริง ๆ ให้กับตัวแปรดังกล่าวนั้นจะทำ

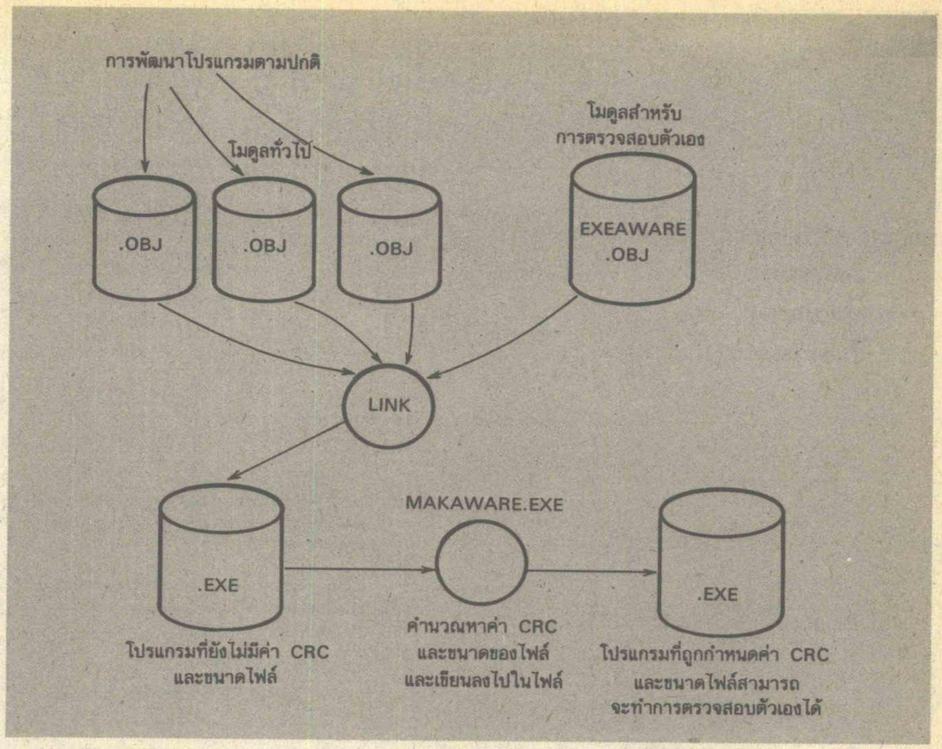
โดยใช้โปรแกรม ซึ่งถูกเขียนแยกทำงานอีกส่วนโดยตั้งชื่อว่า MAKAWARE.EXE จะทำหน้าที่การอ่านเอาไฟล์โปรแกรมที่ถูกเขียนให้สามารถตรวจสอบตัวเองได้ดังกล่าวขึ้นมาคำนวณหาขนาดไฟล์และค่า CRC ของไฟล์ และจากค่าที่ได้ก็จะทำการเขียนลงในไฟล์โปรแกรม ณ.ตำแหน่งของตัวแปรทั้งสองของโมดูล EXEAWARE

คราวนี้เกิดปัญหาขึ้นอีกปัญหาหนึ่ง เราจะทราบที่อยู่หรือตำแหน่งของตัวแปรทั้งสองได้อย่างไร เพื่อที่จะนำเอาค่าที่คำนวณได้เขียนทับลงไปในตำแหน่งที่ถูกต้อง วิธีการแก้ไขปัญหานี้ทำได้ไม่ยากนัก เพื่อให้สามารถค้นหาตำแหน่งที่อยู่ตัวแปรทั้งสองในไฟล์ได้ถูกต้องนั้นเราอาจจะใส่ลายเซ็นต์ (Signature) หรือเป็นข้อความพิเศษไว้ติดกับตัวแปรทั้งสอง (หรืออาจจะใช้ตำแหน่งเดียวกันโดยมองเป็นลักษณะโครงสร้างของยูเนียน) ซึ่งจากข้อความพิเศษดังกล่าวจะทำให้ทราบถึงตำแหน่งตัวแปรทั้งสองได้

การคำนวณหาค่า CRC ของไฟล์โปรแกรมนั้นเพื่อความง่ายของโปรแกรมค่าในไฟล์ทุกไบต์จะถูกนำมาคำนวณยกเว้นค่าในตัวแปรที่ใช้เก็บค่า CRC และขนาดของไฟล์ ขั้นตอนการพัฒนาโปรแกรมให้สามารถตรวจสอบตัวเองได้แสดงเป็นไดอะแกรมไว้ในรูปที่ 1

โมดูลการตรวจสอบตัวเอง

ในโมดูลที่ใช้ตรวจสอบตัวเอง จะมีฟังก์ชันให้เรียกใช้งานเพียงฟังก์ชันเดียวคือ ฟังก์ชัน exeaware ( ) ซึ่งวิธีการใช้แสดงในรูปที่ 2 พารามิเตอร์ที่ส่งผ่านเข้าไปในฟังก์ชันคือ ''pgmname'' นั้น คือชื่อโปรแกรมที่ถูกเรียก


การพัฒนาโปรแกรมตามปกติ
โมดูลทั่วไป
โมดูลสำหรับ
การตรวจสอบตัวเอง
.OBJ
.OBJ
.OBJ
EXEAWARE
.OBJ
LINK
.EXE
MAKAWARE.EXE
.EXE
โปรแกรมที่ยังไม่มีค่า CRC
และขนาดไฟล์
คำนวณหาค่า CRC
และขนาดของไฟล์
และเขียนลงไปในไฟล์
โปรแกรมที่ถูกกำหนดค่า CRC
และขนาดไฟล์สามารถ
จะทำการตรวจสอบตัวเองได้

รูปที่ 1 แสดงขั้นตอนการพัฒนาโปรแกรมให้ตรวจสอบตัวเองได้

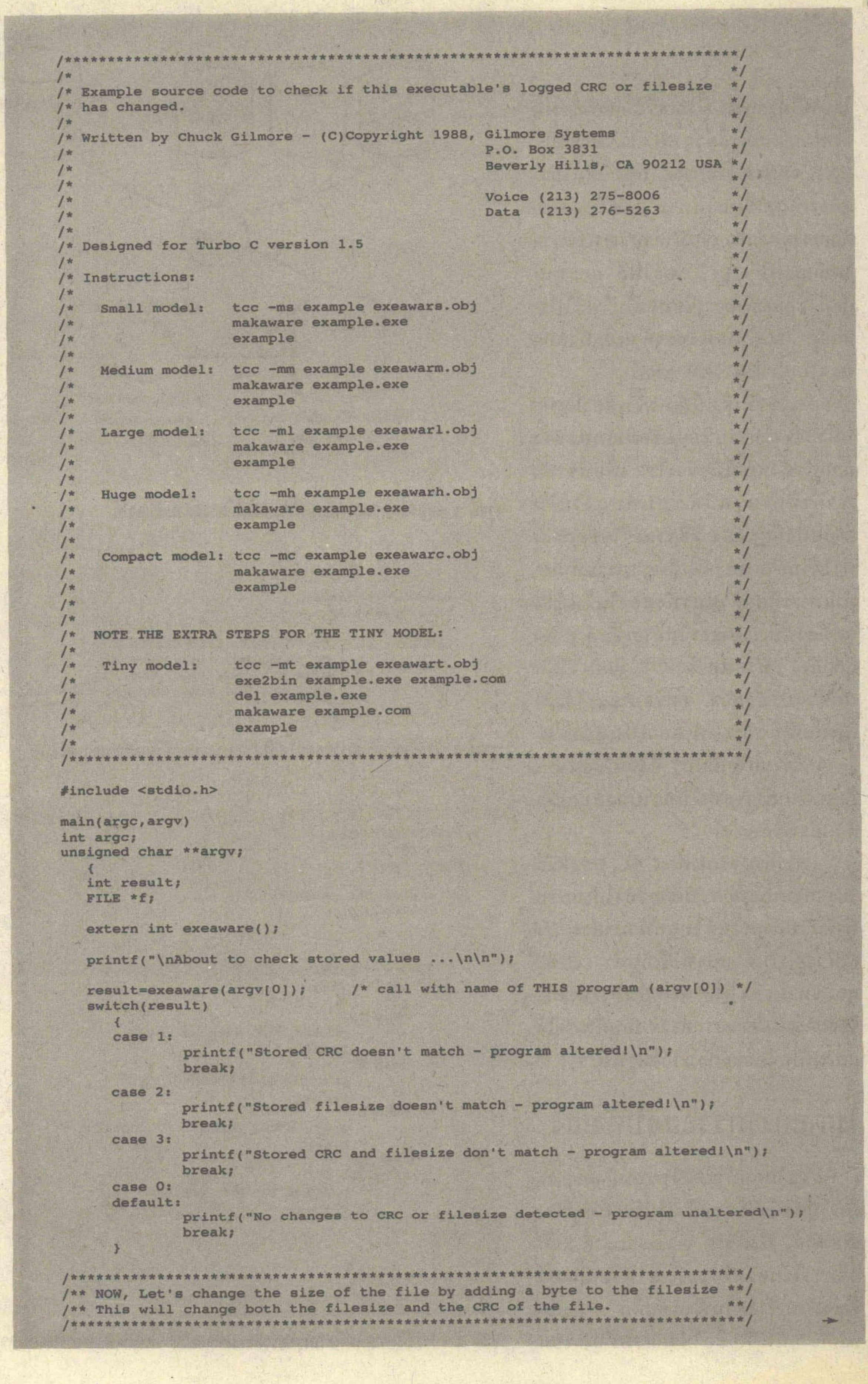
```
exeaware(pgmname)
char *pgmname;

RETURNS:  0 - no changes, file unaltered.
          1 - CRC changed, file altered.
          2 - size changed, file altered.
          3 - both CRC and size changed, file altered.
```

รูปที่ 2 ฟังก์ชันสำหรับตรวจสอบไฟล์โปรแกรมและค่าที่รีเทอร์นจากการทำงาน

```
main(argc,argv)
int argc;
char **argv;
   {
   extern int exeaware();
   int result;
   ;
   ;
   ;
   result = exeaware(argv[0]);  /* this is how to call the function */
   switch(result)
      {
      case 1:
             printf("CRC changed\n");
             break;
      case 2:
             printf("size changed\n");
             break;
      case 3:
             printf("CRC and size changed\n");
             break;
      case 0:
      default:
             printf("No changes\n");
             break;
      }
   ;
   ;
   ;
   }
```

รูปที่ 3 รูปแบบโปรแกรมที่สามารถตรวจสอบตัวเองได้ โดยก่อนการทำงานอื่นใดจะเรียกฟังก์ชัน exeaware ( ) ก่อน



รูปที่ 4 ตัวอย่างโปรแกรมสำหรับทดสอบการทำงานของฟังก์ชัน exeaware ( )

```
/******************************************************************************/
/*                                                                            */
/* Example source code to check if this executable's logged CRC or filesize  */
/* has changed.                                                               */
/*                                                                            */
/* Written by Chuck Gilmore - (C)Copyright 1988, Gilmore Systems              */
/*                                               P.O. Box 3831                */
/*                                               Beverly Hills, CA 90212 USA */
/*                                                                            */
/*                                               Voice (213) 275-8006         */
/*                                               Data  (213) 276-5263         */
/*                                                                            */
/* Designed for Turbo C version 1.5                                           */
/*                                                                            */
/* Instructions:                                                              */
/*                                                                            */
/*   Small model:   tcc -ms example exeawars.obj                              */
/*                  makaware example.exe                                      */
/*                  example                                                   */
/*                                                                            */
/*   Medium model:  tcc -mm example exeawarm.obj                              */
/*                  makaware example.exe                                      */
/*                  example                                                   */
/*                                                                            */
/*   Large model:   tcc -ml example exeawarl.obj                              */
/*                  makaware example.exe                                      */
/*                  example                                                   */
/*                                                                            */
/*   Huge model:    tcc -mh example exeawarh.obj                              */
/*                  makaware example.exe                                      */
/*                  example                                                   */
/*                                                                            */
/*   Compact model: tcc -mc example exeawarc.obj                              */
/*                  makaware example.exe                                      */
/*                  example                                                   */
/*                                                                            */
/*                                                                            */
/*  NOTE THE EXTRA STEPS FOR THE TINY MODEL:                                  */
/*                                                                            */
/*   Tiny model:    tcc -mt example exeawart.obj                              */
/*                  exe2bin example.exe example.com                           */
/*                  del example.exe                                           */
/*                  makaware example.com                                      */
/*                  example                                                   */
/*                                                                            */
/******************************************************************************/

#include <stdio.h>

main(argc,argv)
int argc;
unsigned char **argv;
   {
   int result;
   FILE *f;

   extern int exeaware();

   printf("\nAbout to check stored values ...\n\n");

   result=exeaware(argv[0]);      /* call with name of THIS program (argv[0]) */
   switch(result)
      {
      case 1:
              printf("Stored CRC doesn't match - program altered!\n");
              break;
      case 2:
              printf("Stored filesize doesn't match - program altered!\n");
              break;
      case 3:
              printf("Stored CRC and filesize don't match - program altered!\n");
              break;
      case 0:
      default:
              printf("No changes to CRC or filesize detected - program unaltered\n");
              break;
      }

/******************************************************************************/
/** NOW, Let's change the size of the file by adding a byte to the filesize **/
/** This will change both the filesize and the CRC of the file.             **/
/******************************************************************************/
```

รูปที่ 4 (ต่อ)

```
   printf("about to change filesize\n");
   if(!(f=fopen(argv[0],"ab")))                /* open for "append, binary mode" */
      {
      printf("Error - Can't open \"%s\"\n",argv[0]);
      exit(1);
      }
/*   putc(0,f); */                               /* write out 1 byte */
   fclose(f);                                  /* close the file   */

/****************************************************************************/
/* The following is duplicated code from above (for example purposes only) */
/****************************************************************************/

   result=exeaware(argv[0]);     /* call with name of THIS program (argv[0]) */
   switch(result)
      {
      case 1:
              printf("Stored CRC doesn't match ~ program altered!\n");
              break;
      case 2:
              printf("Stored filesize doesn't match - program altered!\n");
              break;
      case 3:
              printf("Stored CRC and filesize don't match - program altered!\n");
              break;
      case 0:
      default:
              printf("No changes to CRC or filesize detected - program unaltered\n");
              break;
      }

   exit(0);
   }
```

ใช้งานนั้นเอง รูปแบบการเรียกใช้ฟังก์ชันแสดงดังในรูปที่ 3

การเขียนโปรแกรมเพื่อให้สามารถตรวจสอบตัวเองจะต้องทำตามรูปแบบในรูปที่ 3 โดยจะพบว่าก่อนที่โปรแกรมจะทำงานใด ๆ จะเรียกฟังก์ชัน exeaware (argv [0]) ซึ่งพารามิเตอร์ argv [0] นั้นก็คือชื่อของตัวโปรแกรมที่ถูกเรียกทำงานจากดอสหรือโปรแกรมตัวเอง ภายในฟังก์ชัน exeaware ( ) ก็จะทำการตรวจสอบค่า CRC และขนาดของไฟล์แล้วส่งผลลัพธ์ที่ได้จากการเปรียบเทียบตามรูปที่ 2 ซึ่งสามารถจะตรวจสอบแล้วพิมพ์ข้อความบอกเหตุการณ์ที่เกิดขึ้นได้

สำหรับโมดูลที่มีให้ท่านผู้อ่านนำไปใช้งานนั้นได้มาจากบริษัท Gilmore Systems ซึ่งเป็นโมดูลที่ได้รับอนุญาตให้ใช้หรือแจกจ่ายได้ ผู้เขียนก็ขอนำมาฝากให้ท่านผู้อ่านทุก ๆ ท่านที่ต้องการโดยหาได้จาก ซอฟต์เมล์ ในโมดูลดังกล่าวจะใช้หลักการตามที่อธิบายมาตอนต้น และโมดูลที่ให้นั้นเป็นโมดูลหรือ obj ไฟล์สำหรับใช้กับเทอร์โบซีเท่านั้น ซึ่งจะประกอบด้วยไฟล์

exeawart.obj-สำหรับโปรแกรมแบบ tiny model

exeawarc.obj-สำหรับโปรแกรมแบบ Compact model

exeawars.obj-สำหรับโปรแกรมแบบ small model

exeawarm.obj-สำหรับโปรแกรมแบบ medium model

exeawarl.obj-สำหรับโปรแกรมแบบ large model

exeawarh.obj-สำหรับโปรแกรมแบบ huge model

นอกจากนั้นในรูปที่ 4 ได้แสดงไฟล์ตัวอย่าง example.c ซึ่งใช้สำหรับทดสอบการทำงานของโมดูลการตรวจสอบตัวเอง ในส่วนหัวของโปรแกรมตัวอย่างได้แสดงขั้นตอนและวิธีการสร้างโปรแกรมอย่างละเอียด การทำงานในโปรแกรมตัวอย่างนั้นครั้งแรกจะทำการตรวจสอบตัวเองก่อน จากนั้นจะพยายามเปลี่ยนแปลงไฟล์โปรแกรมโดยจะเปิดไฟล์แล้วเขียนค่าศูนย์ลงไปต่อท้ายไฟล์แล้วปิดไฟล์ เมื่อทำการเปลี่ยนแปลงไฟล์เสร็จเรียบร้อยก็จะเรียกฟังก์ชัน exeaware ( ) เพื่อตรวจสอบตัวเองอีกครั้งหนึ่ง

สรุป

ผู้เขียนได้ให้แนวทางการเขียนโปรแกรมเพื่อตรวจสอบตัวเองและโมดูลที่ให้จะสามารถช่วยให้ท่านผู้อ่านพัฒนาโปรแกรมที่มีความสามารถในการตรวจสอบตัวเอง จากการติดเชื้อไวรัสคอมพิวเตอร์ได้